

system either locally (e.g., a direct connection) or remotely (e.g., a remote connection via the internet). Illustratively, port 510 converts the I²C signaling (used on system maintenance bus 7) to the IEEE 1149.1 Joint Task Action Group (JTAG) interface, known in the art. Console 520 is assumed to be an intelligent terminal, such as a personal computer, and provides for administration and maintenance of server 5. (An administration, or maintenance console, is also referred to as a Maintenance Instruction Processor (MIP).) Console 520 stores a log file 505 on a non-volatile memory (like a disk drive) (not shown). Log file 505 is assumed to be an ASCII (American Standard Code for Information Interchange) text file, but can be in any format. Periodically, 10 maintenance processes (or applications) (not shown) executing on console 520 update (or write-to) log file 505 for tracking system events (e.g., a problem (or error)). In this context, and in accordance with a feature of the invention, the above-described controller of the respective boards (e.g., board 200 of FIG. 2) supporting distributed power control in server 5 are used to perform an illustrative maintenance process such as that shown in 15 the flow chart of FIG. 6.

The flow chart of FIG. 6 is similar to the flow chart of FIG. 3 with the addition of steps 320, 325 and 330. Like numbers indicate like steps and are not described further. In step 320, controller 120 of FIG. 2 writes data to log file 505, where the data is representative of the detected problem. (This data is written using the above-mentioned I²C signaling interface, which is then converted to JTAG for transmission to console 520, 20 as noted above. Information exchange, e.g., using I²C signaling, presumes a suitably formatted message set (not shown) for sending commands and receiving status information that may, or may not, include error detection and/or error correction. For example, a message comprises at least three fields, a n bit message field indicating 25 whether the message comprises command or status information, a k bit description field specifying the command or status, and a j bit checksum field.) If the detected problem is a voltage level that is out of range, a record is written to log file 505, the record comprising: a text identifier of the type of problem (e.g. “out of range voltage”); identification of the particular board; the time; and descriptive text indicating whether the 30 voltage was above, or below, the required range. After step 320, execution proceeds back

to step 305 to continue monitoring of the board. With respect to this continued monitoring after a problem was detected, step 325 has been added. If in step 310 a problem is no longer detected, execution proceeds to step 325, where controller 120 checks if a problem was previously detected. (Obviously, suitable state variables (not shown) are set and/or cleared to track this condition. The use of variables to store state information is a known programming technique and not described herein.) If no problem was previously detected execution proceeds to step 305. On the other hand if a problem was previously detected, execution proceeds to step 320, where an indicator that the problem was seemingly corrected is written to log file 505. Since data regarding the health of server 5 is available in log file 505, this data is subsequently accessed by a user from console 520. Similarly, step 330 has been added to keep fault messages from flooding log file 505.

Indeed, the user from console 520 can also, in accordance with the principles of the invention, test server 5 by, e.g., (a) individually varying the voltage on a particular one of those boards supporting distributed power control, and (b) then performing fault diagnostics to examine its effect on the particular board. In this regard, a controller of a board supporting distributing power control (as represented by controller 120 of FIG. 2) receives instructions via the above-mentioned I²C signaling interface. Such a testing method is illustrated by the flow chart of FIG. 7. In step 705, controller 120 receives an instruction (e.g., via console 520 of FIG. 5), where the instruction specifies a particular change to voltage regulator 110. In response, controller 120 adjusts voltage regulator 110 in step 710. Thus, it is possible to run particular tests under different power conditions for those boards of a computer system that support distributed power control.

It should be noted that with the ability for a board supporting distributed power control (e.g., board 200 of FIG. 2) to exchange messages, e.g., via system maintenance bus 7, a shut down of a board can either be sudden and/or graceful. For example, returning for the moment to step 355 of FIG. 4, controller 120 first signals server 5, via the system maintenance bus 7, that board 200 is going to be shutdown and that board 200 should gracefully exit the execution of any pending programs. Either (a) upon receipt 25 from server 5, via system maintenance bus 7, that board 200 has stopped execution, or (b)

the passage of a predefined period of time (i.e., a time-out), controller 120 then may or may not (based on system requirements) perform the shut down of voltage regulator 110.

In addition to those described above, other types of problem detection (or exception handling) - as represented by steps 310 and 350 of FIG. 4 - can occur in accordance with the inventive concept. For example, consider the following. The controller (e.g., controller 120 of FIG. 2) maintains its own history, or data, file. This allows controller 120 to perform time-based analysis of data before (or in addition to) any instantaneous exception reporting. For example, if there is access to temperature sensor data either via signaling path 112 or another signaling path, controller 120 forms an average temperature by accumulating individual temperature value readings over a predefined period of time for storage in memory 185. When this average temperature exceeds a predefined value, controller 120 writes data to a log file (such as log file 505) and/or causes a system alarm to be generated thus, perhaps, predicting the occurrence of a potential problem (e.g., before the board actually fails). Indeed, controller 120 can also shut the board off by disabling voltage regulator 110 as is illustrated by step 355 in the flow chart of FIG. 4.

As another example, the controller performs current shifting analysis, i.e., it tracks current data for the board over time. If the current data begins to increase, this could be suggestive of a pending failure and, in a similar fashion to the above-described shutdown of the board for a temperature failure, shuts down the board when the current exceeds a predefined threshold and/or logs the error to the computer system and/or generates an alert.

Other illustrative embodiments of the invention for use, e.g., in server 5 of FIG. 1, are shown in FIGs. 8 and 9. Other than the inventive concept, the elements shown in these figures are well known and not described in detail. In FIG. 8, a board 800 comprises a power control element as represented by micro-controller 850 and DC-to-DC regulators (or converters) 810 and 805. Board 800 interfaces to the remainder of the system via hot plug control circuit 860 (which, as known in the art, provides the ability to insert and remove board 800 without turning off power to other parts of the system). (It should be noted that the ability to hot plug a board can also be used on the illustrative